

Human Computer Interaction CS449 – CS549

Assignment 5: Development of Gesture-based interaction using Mediapipe (Draft, subject to change)

Grade: 10 out of 100

Due Date: December 9th Monday

In this assignment, you will implement a **gesture recognition** based interactive system by using the **MediaPipe library**. You must create gestures using either face landmarks, hand landmarks, or body pose landmarks provided in this document. You are required to use a real-time stream from your laptop's or mobile's **camera** to detect the gestures. Once you define your gestures, they must trigger an interaction or event. This interaction can be either **PC-based** or mobile-based.

After implementing your system, in the next assignment (Assignment-6) you will conduct a usability test with users.

Part 1: Gesture Options

You can choose to create gestures in one of the following categories (you may use more than one gesture if you want):

1. **Face Gesture:**

Use MediaPipe's face landmark detection to create a custom facial gesture (e.g., a wink, eyebrow raise).

https://ai.google.dev/edge/mediapipe/solutions/vision/face_landmarker

2. **Hand Gesture:**

Use MediaPipe's hand landmark detection to create a custom hand gesture (e.g., a wave, thumbs-up).

https://ai.google.dev/edge/mediapipe/solutions/vision/gesture_recognizer

3. **Body Pose Gesture:**

Use MediaPipe's pose detection to create a custom body gesture (e.g., a specific pose like a yoga move or arm raise).

https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker

Part 2: Interaction Requirement

Each gesture you create must trigger an interaction or event in your system. The interaction can either be on your computer or mobile device. Be creative with what actions your gestures trigger.

Interface Requirements:

1. **Gesture-Based Cursor Control:**

- Implement a cursor that can be controlled by hand, face, or body gestures.

- The cursor should hover over interactive elements such as buttons, scrollbars, or other components.

2. Gestures for Different Actions:

- Assign distinct gestures for different actions (e.g., clicking, scrolling, dragging).
 - **Examples:**
 - Hand gesture (e.g., pinching) to select or click a button.
 - Face gesture (e.g., blinking) to confirm an action.
 - Body pose (e.g., raising an arm) to scroll content or navigate.

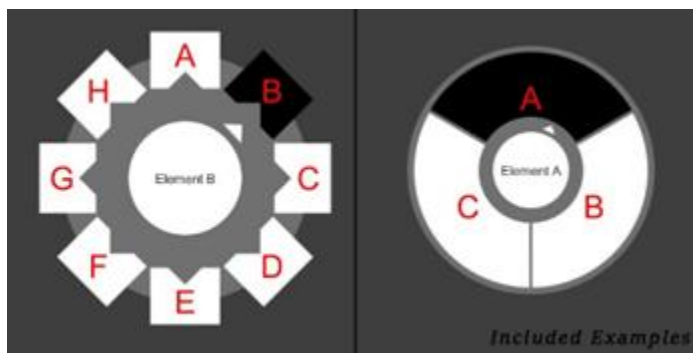
3. Visual Feedback:

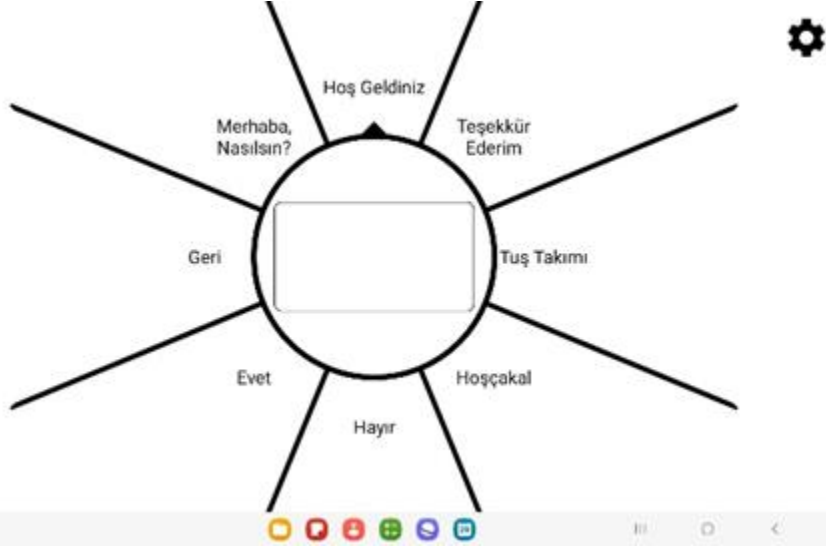
- Provide visual feedback to indicate when a gesture is detected or an action is being performed.
 - Change the **color** of elements when the cursor hovers over them.
 - Highlight buttons or scrollbars when they are selected or clicked.
 - Use additional visual cues, such as subtle animations or effects, to improve user experience.

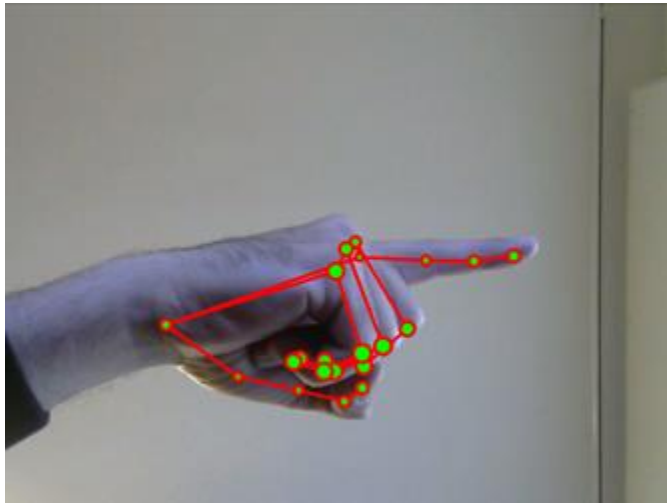
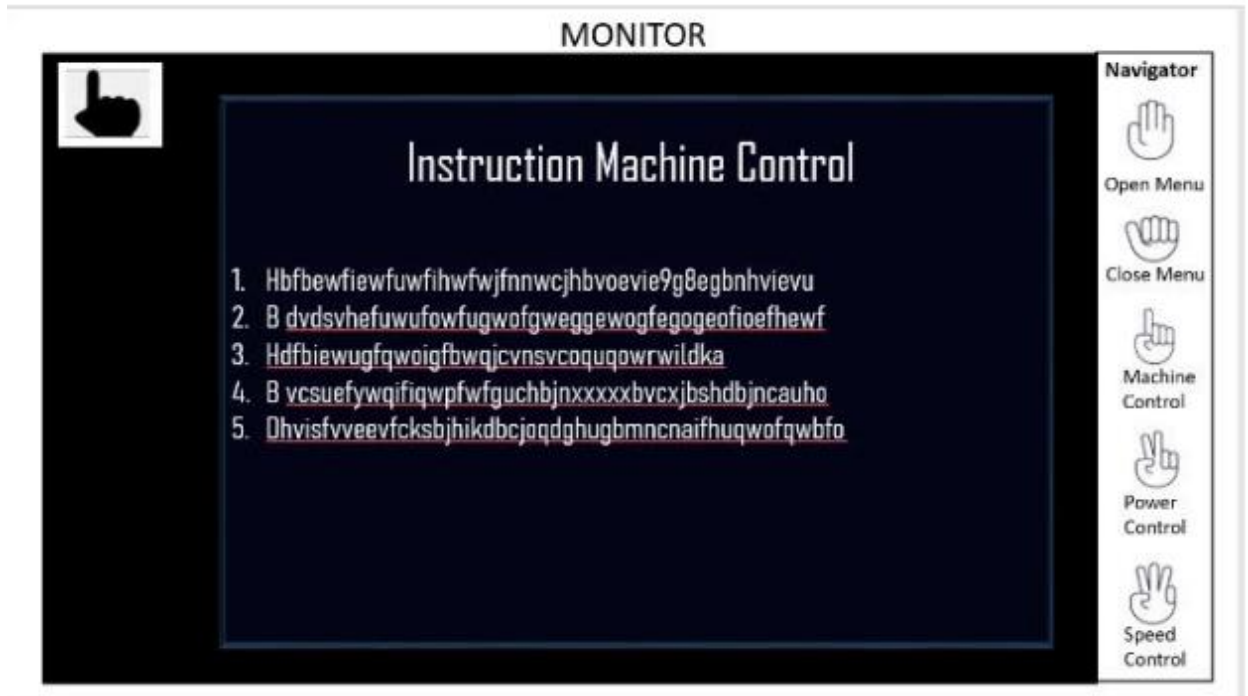
4. Components of the Interface:

- **Scrollbars (Vertical and Horizontal):**
 - The interface should include both vertical and horizontal scrollbars.
- **Clickable Buttons:**
 1. Include buttons that can be clicked or selected using a hand, face, or body gesture.

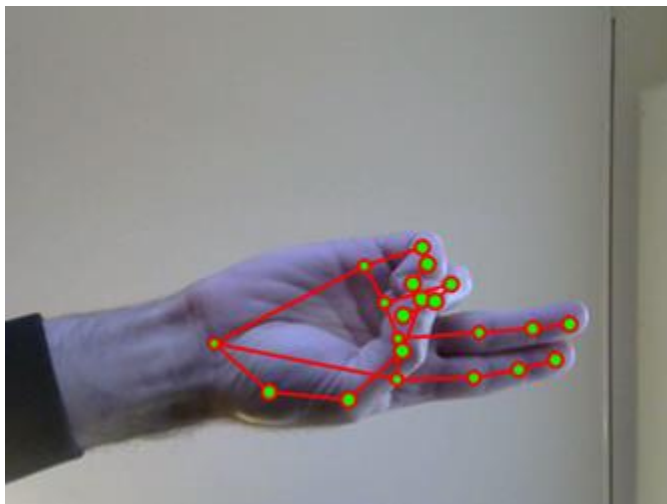
5. Example interfaces and Gestures:







Cursor-pointing Gesture.



Scrolling Gesture

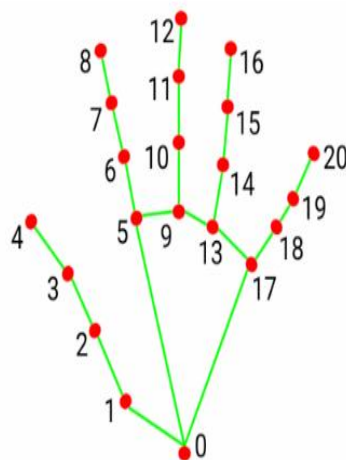


Reload Gesture

Part 3: Implementation

1. Install MediaPipe and any other required libraries such as opencv.
2. Choose and calculate the landmarks for your gesture. (Provide visual illustrations and documentation on how to calculate the landmarks based on their position).

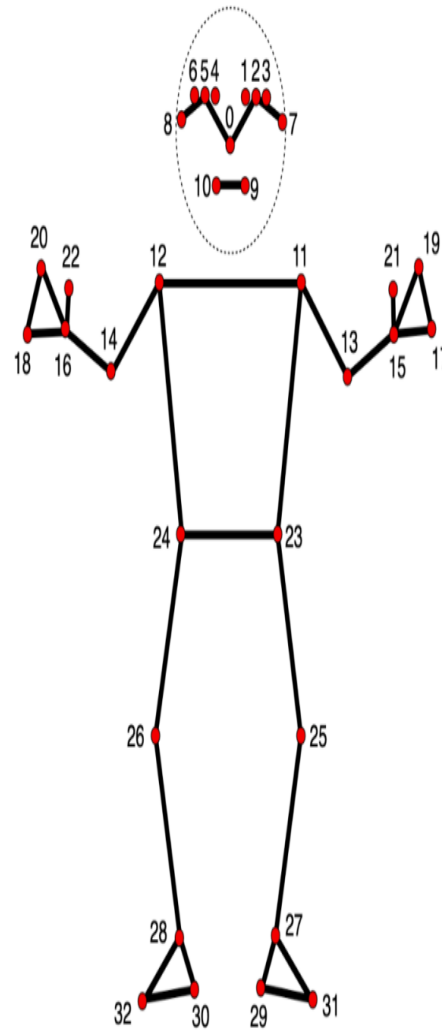
a. Hand landmarks:



0. WRIST	11. MIDDLE_FINGER_DIP
1. THUMB_CMC	12. MIDDLE_FINGER_TIP
2. THUMB_MCP	13. RING_FINGER_MCP
3. THUMB_IP	14. RING_FINGER_PIP
4. THUMB_TIP	15. RING_FINGER_DIP
5. INDEX_FINGER_MCP	16. RING_FINGER_TIP
6. INDEX_FINGER_PIP	17. PINKY_MCP
7. INDEX_FINGER_DIP	18. PINKY_PIP
8. INDEX_FINGER_TIP	19. PINKY_DIP
9. MIDDLE_FINGER_MCP	20. PINKY_TIP
10. MIDDLE_FINGER_PIP	

- b. Face mesh landmarks: <https://github.com/tensorflow/tfjs-models/blob/838611c02f51159afdd77469ce67f0e26b7bbb23/face-landmarks-detection/src/mediapipe-facemesh/keypoints.ts>
- c. Body landmarks:

0 - nose
 1 - left eye (inner)
 2 - left eye
 3 - left eye (outer)
 4 - right eye (inner)
 5 - right eye
 6 - right eye (outer)
 7 - left ear
 8 - right ear
 9 - mouth (left)
 10 - mouth (right)
 11 - left shoulder
 12 - right shoulder
 13 - left elbow
 14 - right elbow
 15 - left wrist
 16 - right wrist
 17 - left pinky
 18 - right pinky
 19 - left index
 20 - right index
 21 - left thumb
 22 - right thumb
 23 - left hip
 24 - right hip
 25 - left knee
 26 - right knee
 27 - left ankle
 28 - right ankle
 29 - left heel
 30 - right heel
 31 - left foot index
 32 - right foot index



3. Write a program to run on your PC or Mobile device to recognize your gesture and trigger an event.
4. Ensure that the program runs efficiently and can detect the gesture in real time.

Part 4: Documentation and Submission

You must provide documentation that outlines:

1. **Overview of the system:** Explain the gesture chosen, the landmarks used, and the interaction it triggers.
2. **Implementation Details:** Provide a brief explanation of how you calculated the landmarks and wrote the code.
3. **Code:** Include the code in a well-organized format.
4. **Github:** Since this is a group project, you must use GitHub and share your repository to track the contribution of each member. Explain group members' amount of contribution to software development in your report by referring to Github data.

5. **Video Demo:** Submit a short video demonstrating your system in action. (We may ask groups to make a face to face demo)

Grading

Working Software – 75 pts

Project Report - 20 pts

Peer Evaluation: 5 pts **This will be requested from each group member separately and the grades will be confidential.**

Warning: Depending on Github results and peer evaluation, students' individual grades might be different.